

- (2) assigning a unique one of the plurality of index-addressable temporary storage locations to each one of said plurality of instructions in said instruction buffer, wherein an output corresponding to a given one of said plurality of instructions is stored in said index-addressable temporary storage location assigned to said given one of said plurality of instructions;
- (3) determining whether one of said plurality of instructions in said instruction buffer is a dependent instruction, wherein said dependent instruction has an input that is an output of a previous instruction, wherein said previous instruction is an instruction in said instruction buffer that precedes said dependent instruction in the prescribed program order; and
- (4) associating said index-addressable temporary storage location assigned to said previous instruction with said input.

- ²
13. The method of claim ¹12, further comprising the steps of:
- (5) executing said dependent instruction only after said previous instruction produces an output;
- (6) storing said output in said index-addressable temporary storage location assigned to said previous instruction; and
- (7) performing an operation corresponding to said dependent instruction using said output stored in said index-addressable temporary storage location assigned to said previous instruction as said input.

- ³
14. The method of claim ¹12, further comprising the step of associating a done signal with said input, wherein said done signal indicates a status of said previous instruction.

- ⁴
15. The method of claim ¹12, further comprising the step of storing said output in an appropriate register when said previous instruction is retired.

- ⁵
16. A ^{superscalar} ~~superscalar~~ processor for executing instructions having a prescribed program order, comprising:
- an instruction buffer for storing a plurality of instructions;

4 an index-addressable temporary buffer comprising a plurality of temporary storage
5 locations, wherein each one of said plurality of instructions is assigned to a unique one of said
6 plurality of temporary storage locations, wherein an output corresponding to a given one of said
7 plurality of instructions is stored in said temporary storage location assigned to said given one
8 of said plurality of instructions;

9 a data dependency checker to locate a dependent instruction stored in said instruction
10 buffer, wherein said dependent instruction has an input that is dependent on a previous instruction,
11 wherein said previous instruction is an instruction in said instruction buffer that precedes said
12 dependent instruction in the prescribed program order; and

13 a circuit that receives from said data dependency checker dependency data corresponding
14 to said dependent instruction and uses said dependency data to associate said temporary storage
15 location assigned to said previous instruction with said input.

C B1
6 5
17. The ^{superscalar} ~~superscalar~~ processor of claim 16, wherein said circuit associates said temporary
2 storage location assigned to said previous instruction with said input by outputting a reference
3 corresponding to said temporary storage location assigned to said previous instruction.

C 7 6
18. The ^{superscalar} ~~superscalar~~ processor of claim 17, wherein said reference comprises an address.

C 8 7
19. The ^{superscalar} ~~superscalar~~ process of claim 18, wherein said reference further comprises a 1-bit
2 identifier that indicates whether said address is an address of one of said temporary storage
3 locations or is an address of a register.

C 9 7
20. The ^{superscalar} ~~superscalar~~ processor of claim 18, wherein said circuit further outputs a first signal
2 indicating a completion status of said previous instruction.

C 10 9
21. The ^{superscalar} ~~superscalar~~ processor of claim 20, further comprising:
2 a multiplexer having a plurality of inputs, wherein said multiplexer receives an address of
3 a storage location at each one of said plurality of inputs, wherein one of said addresses received
4 at one of said plurality of inputs is said address included in said reference outputted by said circuit;
5 and

an issuer that receives as an input said signal indicating a completion status of said previous instruction and that outputs a select signal to said multiplexer.

11
22. The ^{superscalar} ~~superscalar~~ processor of claim 16⁵, wherein said circuit comprises a priority encoder and a multiplexer, wherein said encoder receives said dependency data, and an output of said encoder is used as a select signal for said multiplexer.

12
23. The ^{superscalar} ~~superscalar~~ processor of claim 22¹¹, wherein said multiplexer receives a plurality of references as inputs and outputs a reference corresponding to said output of said encoder, and wherein said reference outputted by said multiplexer represents an address of said temporary storage location assigned to said previous instruction.

13
24. The ^{superscalar} ~~superscalar~~ processor of claim 16⁵, wherein said instruction buffer is capable of storing at most X number of instructions, and said temporary buffer includes at least X number of temporary storage locations, wherein X is a positive integer.

14
25. The ^{superscalar} ~~superscalar~~ processor of claim 16⁵, wherein one of said plurality of instructions is assigned to a unique one of said plurality of storage locations based on a position of said one of said plurality of instructions within said instruction buffer.

15
26. The ^{superscalar} ~~superscalar~~ processor of claim 16⁵, wherein said data dependency checker locates a dependent instruction stored in said instruction buffer by comparing a source register of one of said plurality of instructions to a destination register of each instruction in said instruction buffer that precedes said one of said plurality of instructions in the prescribed program order.

16
27. The ^{superscalar} ~~superscalar~~ processor of claim 15¹⁵, wherein said data dependency checker comprises a plurality of data dependency circuits, wherein each dependency circuit performs at least on comparison to determine whether any given one of said plurality of instructions depends on a previous instruction.

36

C 1 ¹⁷
28. The ~~superscalar~~ ^{superscalar} processor of claim ¹⁶27, wherein one of said data dependency circuits
2 comprises at least one comparator circuit, said comparator circuit receiving a first input and a
3 second input and outputting a dependency signal, said first input corresponding to a source
4 register of a first instruction stored in said instruction buffer and said second input corresponding
5 to a destination register of a second instruction stored in said instruction buffer, wherein said
6 second instruction precedes said first instruction in the prescribed program order, and wherein
7 said dependency signal indicates whether said first instruction is dependent on said second
8 instruction.

C 1 ¹⁸
29. The ~~superscalar~~ ^{superscalar} processor of claim ¹⁷28, wherein said comparator circuit comprises a
2 plurality of exclusive NOR gates (XNOR) and an AND gate having a plurality of inputs, wherein
3 an output of each XNOR gate is tied to an input of said AND gate, and wherein each XNOR has
4 a first XNOR input and a second XNOR input, wherein each first XNOR input is tied to a bit of
5 said first input and each second XNOR input is tied to a bit of said second input.

1 ¹⁹
30. A computer system, comprising:
2 a memory unit for storing program instructions having a prescribed program order;
3 a bus for retrieving said program instructions from said memory unit; and
4 a processor in communication with said bus for executing said program instructions,
5 wherein said processor comprises:
6 an instruction buffer for storing a plurality of instructions;
7 an index-addressable temporary buffer comprising a plurality of temporary storage
8 locations, wherein each one of said plurality of instructions is assigned to a unique one of
9 said plurality of temporary storage locations, wherein an output corresponding to a given
10 one of said plurality of instructions is stored in said temporary storage location assigned
11 to said given one of said plurality of instructions;
12 a data dependency checker to locate a dependent instruction stored in said
13 instruction buffer, wherein said dependent instruction has an input that is dependent on
14 a previous instruction, wherein said previous instruction is an instruction in said
15 instruction buffer that precedes said dependent instruction in the prescribed program
16 order; and

37

17 a circuit that receives from said data dependency checker dependency data
18 corresponding to said dependent instruction and uses said dependency data to associate
19 said temporary storage location assigned to said previous instruction with said input.

1 ²⁰
~~31.~~ The computer system of claim ¹⁹~~30~~, wherein said circuit associates said temporary storage
2 location assigned to said previous instruction with said input by outputting an address of said
3 temporary storage location assigned to said previous instruction.

1 ²¹
~~32.~~ The computer system of claim ²⁰~~31~~, wherein said circuit further outputs a first signal
2 indicating a completion status of said previous instruction.

1 ²²
~~33.~~ The computer system of claim ²¹~~32~~, further comprising:
2 a multiplexer having a first plurality of inputs, wherein said multiplexer receives an address
3 of a storage location at each one of said plurality of inputs, wherein one of said addresses received
4 at one of said plurality of inputs is said address outputted by said circuit; and
5 an issuer that receives as an input said signal indicating a completion status of said
6 previous instruction, and that outputs a select signal to said multiplexer.

1 ²³
~~34.~~ The computer system of claim ¹⁹~~30~~, wherein said circuit comprises a priority encoder and
2 a multiplexer, wherein said encoder receives said dependency data, and an output of said encoder
3 is used as a select signal for said multiplexer.

1 ²⁴
~~35.~~ The computer system of claim ²³~~34~~, wherein said output of said encoder represents said
2 previous instruction.

1 ²⁵
~~36.~~ The computer system of claim ²⁴~~35~~, wherein said multiplexer receives a plurality of
2 references as inputs and outputs a reference corresponding to said output of said encoder, wherein
3 said reference outputted by said multiplexer represents an address of said temporary storage
4 location assigned to said previous instruction.

38

²⁴¹⁹

1 ²⁴~~37~~. The computer system of claim ¹⁹~~30~~, wherein said instruction buffer is capable of storing at
2 most X number of instructions, and said temporary buffer includes at least X number of temporary
3 storage locations, wherein X is a positive integer.

²⁷¹⁹

1 ²⁷~~38~~. The computer system of claim ¹⁹~~30~~, wherein one of said plurality of said program
2 instructions is assigned to a unique one of said plurality of storage locations based on a position
3 of said one of said plurality of said program instructions within said instruction buffer.

²⁸¹⁹

1 ²⁸~~39~~. The computer system of claim ¹⁹~~30~~, wherein said data dependency checker locates a
2 dependent instruction stored in said instruction buffer by comparing a source register of one of
3 said plurality of said program instructions to a destination register of each instruction in said
4 instruction buffer that precedes said one of said plurality of said program instructions in the
prescribed program order.

²⁹²⁸

1 ²⁹~~40~~. The computer system of claim ²⁸~~39~~, wherein said data dependency checker comprises a
2 plurality of data dependency circuits, wherein each dependency circuit performs at least on
3 comparison to determine whether any given one of said plurality of said program instructions
4 depends on a previous instruction.

³⁰²⁹

1 ³⁰~~41~~. The computer system of claim ²⁹~~40~~, wherein one of said data dependency circuits comprises
2 at least one comparator circuit, said comparator circuit receiving a first input and a second input
3 and outputting a dependency signal, said first input corresponding to a source register of a first
4 instruction stored in said instruction buffer and said second input corresponding to a destination
5 register of a second instruction stored in said instruction buffer, wherein said second instruction
6 precedes said first instruction in the prescribed program order, and wherein said dependency
7 signal indicates whether said first instruction is dependent on said second instruction.

³¹³⁰

1 ³¹~~42~~. The computer system of claim ³⁰~~41~~, wherein said comparator circuit comprises a plurality
2 of exclusive NOR gates (XNOR) and an AND gate having a plurality of inputs, wherein an output
3 of each XNOR gate is tied to an input of said AND gate, and wherein each XNOR has a first

39

4 XNOR input and a second XNOR input, wherein each first XNOR input is tied to a bit of said
5 first input and each second XNOR input is tied to a bit of said second input.

C 1 ³²
23. A ~~superscalar~~ ^{superscalar} processor for executing instructions having a prescribed program order,
2 comprising:

3 an instruction buffer storing a plurality of instructions;
4 a register file having a plurality of registers and a plurality of index-addressable temporary
5 storage locations, wherein each one of said plurality of instructions is assigned to a unique one
6 of said plurality of temporary storage locations and one of said plurality of registers, wherein an
7 output corresponding to a given one of said plurality of instructions is stored in said temporary
8 storage location assigned to said given one of said plurality of instructions;

9 a data dependency checker to locate dependent instructions stored in said instruction
10 buffer, wherein a dependent instruction is an instruction that should not be executed until after
11 a particular previous instruction within said instruction buffer is executed, wherein said data
12 dependency checker outputs dependency information corresponding to each instruction stored
13 in said instruction buffer;

14 a plurality of circuits, wherein each of said plurality of circuits corresponds to an
15 instruction in said instruction buffer and receives dependency data from said data dependency
16 checker, wherein said dependency data received at a particular circuit corresponds to said
17 instruction to which said particular circuit corresponds, and wherein a circuit corresponding to
18 a given dependent instruction outputs a reference representing said temporary storage location
19 assigned to a particular previous instruction that must be executed prior to said given dependent
20 instruction.

1 ³³
44. A computer system, comprising:
2 a memory unit for storing program instructions having a prescribed program order;
3 a bus for retrieving said program instructions from said memory unit; and
4 a processor in communication with said bus for executing said program instructions,
5 wherein said processor comprises:

6 an instruction buffer storing a plurality of instructions;

7 a register file having a plurality of registers and a plurality of index-addressable
8 temporary storage locations, wherein each one of said plurality of instructions is assigned
9 to a unique one of said plurality of temporary storage locations and one of said plurality
10 of registers, wherein an output corresponding to a given one of said plurality of
11 instructions is stored in said temporary storage location assigned to said given one of said
12 plurality of instructions;

13 a data dependency checker to locate dependent instructions stored in said
14 instruction buffer, wherein a dependent instruction is an instruction that should not be
15 executed until after a particular previous instruction within said instruction buffer is
16 executed, wherein said data dependency checker outputs dependency information
17 corresponding to each instruction stored in said instruction buffer; and

18 a plurality of circuits, wherein each of said plurality of circuits corresponds to an
19 instruction in said instruction buffer and receives dependency data from said data
20 dependency checker, wherein said dependency data received at a particular circuit
21 corresponds to said instruction to which said particular circuit corresponds, and wherein
22 a circuit corresponding to a given dependent instruction outputs a reference representing
23 said temporary storage location assigned to a particular previous instruction that must be
24 executed prior to said given dependent instruction.

34
35. 1 In a computer system having a register file comprising a plurality of registers and a
2 plurality of index-addressable temporary storage locations, a method for executing instructions
3 having an input and an output and having a prescribed program order, comprising the steps of:

4 (1) assigning a unique one of the plurality of index-addressable temporary storage
5 locations to each one of a plurality of instructions in an instruction buffer, wherein the output
6 corresponding to a given one of said plurality of instructions is stored in said temporary storage
7 location assigned to said given one of said plurality of instructions;

8 (2) determining whether one of said plurality of instructions in said instruction buffer
9 is a dependent instruction, wherein said dependent instruction has an input that is dependent on
10 a previous instruction, wherein said previous instruction is an instruction in said instruction buffer
11 that precedes said dependent instruction in the prescribed program order; and